Distributed event-triggered scheduling of wireless networked control systems

Adam Molin[†*], Sandra Hirche[†]

[†] Institute of Automatic Control Engineering, Technische Universität München, Germany, http://www.lsr.ei.tum.de

[*]Corresponding Author: adam.molin@tum.de

Keywords: networked control systems, control over communications, event-triggered sampling, medium access control.

Computer control systems are commonly designed in such way that observations are sampled equidistantly in time (time-triggered sampling). Networked control systems (NCS) are a special class of computer controlled systems, where sensors, controllers and actuators are spatially distributed and are only able to exchange information over a digital communication network. Due to the fact that these communicating devices are sharing a common medium, one of the major concerns of NCS is to minimize communication load, while preserving a certain level of control performance. It has been shown in [2] for multi-loop NCS that event-triggered sampling can be much more efficient than time-triggered sampling schemes in terms of resource usage of the communication system.

While this approach presumes reliable communication systems, the focus of this work is to consider event-triggered scheduling schemes in unreliable communication networks. In [2], packet collisions in CSMA (carrier sense multiple access) protocols are resolved through arbitration mechanisms. If multiple senders access the shared medium at the same time, one sender will be elected to transmit information to its destination. In the absence of a global coordinator, arbitration mechanisms are hard to sustain in wireless communication networks [4]. We therefore consider in the following multi-loop NCS in CSMA communication networks without an arbitration mechanism, i.e. in case of multiple access to the medium at the same time collision occurs and all packets are lost.

The NCS is given by N control loops which are closed over a common communication network. Each process i to be controlled is given by a stochastic linear discrete-time scalar system

$$x_i[k+1] = a_i x_i[k] + b_i u_i[k] + w_i[k], \quad x_i[0] = 0, \quad i = 1, \ldots N, \tag{1}$$

$$y_i[k] = \prod_{j \neq i} (1 - \delta_j[k]) \delta_i[k] x_i[k] \tag{2}$$

where b_i is non-zero and the system noise w_i is independent and identically distributed Gaussian noise with zero-mean. The noise sequences are assumed to be mutually statistically independent. Admissible control and scheduling policies are given by

$$u_i[k] = g_k^i(y_i[0], \ldots, y_i[k]), \qquad \delta_i[k] = f_k^i(\mathcal{I}_k) = \begin{cases} 1 & \text{measurement } x_i[k] \text{ sent} \\ 0 & \text{no measurement transmitted} \end{cases} \tag{3}$$

We will consider two different information patterns for the scheduling policy: a global information pattern having complete state information and a distributed information pattern having only local state information.

The objective is to design optimal g_k^i and f_k^i that minimize the aggregate cost $J = \sum_{i=1}^N J_i$ with

$$J_i = \limsup_{T \to \infty} \frac{1}{T} \mathsf{E}[\sum_{k=1}^{T} Q x_i^2 + R u_i^2], \quad R, Q > 0. \tag{4}$$

Under the assumption of having a global scheduler, it can be shown similarly to [3] that the problem can be separated into three subproblems: a linear quadratic regulator (LQR) problem, an estimation problem and a scheduling problem. The first two can be solved by standard methods of stochastic optimal control [1]. The scheduling problem can be written in the following form:

$$\min_{f_k^i} \limsup_{T\to\infty} \sum_{i=1}^{N} \frac{1}{T} \mathsf{E}[\sum_{k=1}^{T} \Big(1 - \prod_{j\neq i}(1-\delta_j[k])\delta_i[k]\Big)\gamma_i e_i^2[k]]$$

$$e_i[k+1] = \Big(1 - \prod_{j\neq i}(1-\delta_j[k])\delta_i[k]\Big)a_i e_i[k] + w_i[k], \quad e_i[0] = 0 \quad i = 1,\ldots,N,$$

where γ_i is a positive value computed by the system and cost function parameters and can be viewed as a measure of priority of feedback loop i. The parameter $e_i[k]$ is the one-step predicted estimation error at the controller. In case of successful transmission of process i, the error is reset to zero and the one-step predicted error becomes $e_i[k] = w_i[k]$. When having a global coordinator, i.e. each $\delta_i[k]$ may be a function of $e_k = [e_1[k],\ldots,e_N[k]]^T$, the optimal solution can be computed by the dynamic programming algorithm. The solution will assign for each state e_k a particular process to transmit its state update to the controller, i.e. collisions will not occur for such scheme.

As already mentioned, a global coordinator is difficult to realize in wireless networks. Therefore, we draw our attention to distributed scheduling schemes in the following. Although no guarantee can be given that the initial problem of minimizing J can be separated into three subproblems without losing optimality alike having a global coordinator, we assume the same controller and estimator design for the distributed scheduling scheme as for the global scheme. We restrict the scheduling policy $\delta_i[k]$ being a threshold function of $|e_i[k]|$, i.e. send information if $|e_i[k]|$ exceeds a threshold. In addition the scheduler is assumed to be acknowledged whether information has been transmitted successfully. This assumption is very important, as it enables (i) the scheduler to compute $e_i[k]$ and (ii) implement congestion control. In the following we give a statement about the requirements on the distributed scheduling scheme in order to cater for stability. Stability is defined as having bounded moment of the state e_k, when $k \to \infty$.

Lemma 1 *If the scheduling policy has a fixed policy and at least two processes have* $|a_i| > 1$*, then the aggregate system is unstable.*

The proof is omitted due to page limitations. To avoid collisions and achieve stability of the aggregate system, one can think about two possibilities: (i) increasing the threshold, when collisions occur or (ii) choose a random time the process has to wait after a collision until the process can transmit again. The latter approach is the common approach in CSMA access schemes [4]. In fact it turns out that one can construct examples, where the first possibility fails, as the unstable processes may synchronize to each other trying to retransmit its state at same times. Hence, it can be conjectured, that a distributed scheduling policies require additionally some randomized mechanism to resolve collisions. Numerical simulations show that using a threshold policy including some randomness outperforms standard randomized CSMA schemes in terms of aggregate control performance.

For distributed scheduling problems in NCS it seems to be promising to combine randomized and event-triggered scheduling approaches in order to increase control performance, while guaranteeing stability of the communication network and the physical processes.

References

[1] K. J. Åström. *Introduction to Stochastic Control Theory.* Dover Publications, 2006.

[2] A. Cervin and T. Henningsson. Scheduling of event-triggered controllers on a shared network. In *Proc. 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2008.

[3] A. Molin and S. Hirche. On LQG joint optimal scheduling and control under communication constraints. In *Proc. 48th IEEE Conference on Decision and Control*, Shanghai, China, Dec. 2009. Accepted.

[4] A. S. Tanenbaum. *Computer Networks.* Prentice Hall, 2002.